Social and Spatial Proactive Caching for Mobile Data Offloading

Ejder Bastug, Mehdi Bennis, Mérouane Debbah

Social and Spatial Proactive Caching for Mobile Data Offloading

Ejder Baştuğ[◇], Mehdi Bennis[⋆] and Mérouane Debbah[◇],
[◇] Alcatel-Lucent Chair - SUPÉLEC, Gif-sur-Yvette, France
[⋆] Centre for Wireless Communications, University of Oulu, Finland
{ejder.bastug, merouane.debbah}@supelec.fr, bennis@ee.oulu.fi

***Abstract*—The surge in video traffic and shift toward on-demand content consumption is straining mobile operators' networks to a breaking point. In this article, we investigate the problem of mobile data offloading for beyond 4G networks from a *caching* perspective. Leveraging notions of prediction, storage, and social networking, it is shown that peak traffic demands can be substantially reduced by proactively serving predictable user demands, through caching at the network edge (i.e., base stations and users' devices). Notably, we focus on two caching scenarios which exploit the spatial and social structure of the network. Firstly, in order to alleviate backhaul congestion, we propose a mechanism whereby files are proactively cached during off-peak demands based on file popularity and correlations among users-files patterns. Secondly, leveraging social networks and device-to-device (D2D) communications, we propose a procedure that exploits the social structure of the network by predicting the set of influential users to cache strategic contents and disseminate them among their social ties. Numerical results show that important gains are incurred, with backhaul savings and a higher ratio of satisfied users of up to 22% and 26%, respectively. Higher gains can further be obtained by increasing the storage capability at the network edge.**

***Index Terms*—5G, small cell networks, proactive caching, backhaul offloading, D2D communications, social networks.**

I. Introduction

The rapid proliferation of smartphones has substantially enriched the mobile experience, leading to new wireless services, including multimedia streaming, web-browsing applications and socially-interconnected networks. Currently, mobile video streaming accounts for 50% of mobile data traffic and is expected to have a 500-fold increase over the next ten years [2]. At the same time, online social networking (Facebook, Twitter, Digg, etc.) is the second largest contributor to this traffic with a 15% average share [3]. These new phenomena compel mobile operators to redesign their networks and seek more advanced techniques in a cost-effective manner.

One way of taming these unrelenting demands is via the deployment of small cell networks (SCNs) [4], [5], by deploying short-range, low-power, and low-cost small base stations (SBSs) underlaying the macrocellular network. The gist of SCN studies revolve around self-organization, inter-cell interference coordination (ICIC), traffic offloading [6], energy-efficiency, etc (see [5] and references therein). These studies are based on the classical networking paradigm, referred to as *reactive*, in which users' requests are immediately served upon arrival or dropped causing outages. Hence, in order to cater for peak traffic demands expensive high-speed backhaul deployments are required, leading to substantial operational expenditures (OPEX). These key observations mandate a *novel* networking paradigm taking into account recent advances in storage, context-awareness, and social networking [7].

The novel networking paradigm is *proactive* in that network nodes (i.e., base stations and handhelds/smartphones) exploit users' context information and predict users' demands to satisfy their quality-of-service (QoS) requirements [8]. This paradigm goes beyond current cellular networks which have been designed under the precepts of *dumb* user terminals with limited storage and processing capabilities. Exploiting the smartness of these sophisticated devices can substantially enhance the way contents are predicted before users actually request them by storing them at the network edge[1]. As a result, significant resource savings are achieved, minimizing operational and capital expenditures [5].

The idea of caching has recently received tremendous attention. In [9], the idea of femtocaching was proposed in which BSs have low-bandwidth backhaul links and high storage capabilities. [10] explored the notion of proactive resource allocation exploiting the predictability of user behavior for load balancing. The scaling law of the outage probability is derived as a function of a prediction time window using large deviation theory. In a similar vein, [11] studied the asymptotic scaling laws of caching in device-to-device (D2D) in which users collaborate by caching popular content. Nevertheless, while interesting, these works do not address the dynamics of proactive caching under uncertainty, overlooking aspects of context-awareness and social networks. This article aims at filling the void in the dynamics of proactive network caching.

Our key observation is that given the vast amount of information often available, and the fact that human behavior can be predicted, users' future requests can be inferred upon [12]. In this paper, we propose a proactive caching framework leveraging context-awareness and storage constraints at the network edge to alleviate peak data demands and offload traffic. Specifically, by exploiting the predictability of future demands, popular contents are proactively cached before users

This research has been supported by the ERC Starting Grant 305123 MORE (Advanced Mathematical Tools for Complex Network Engineering) and the SHARING project under the Finland grant 128010.

[1]Network edge refers to both small cell base stations (SBSs) and user terminals (UTs).

actually request them. Further, whenever D2D communication is possible, the proposed caching approach exploits users' social ties (relationships and influences within their social community), physical proximity and users' storage for content dissemination.

This paper is structured as follows. The system model and problem formulation of both caching scenarios are presented in Section II. Numerical results are given in Section III, and the impact of various parameters of interest on the figures of merits are discussed. We finally conclude in Section IV.

II. Problem Formulation

Consider a scenario formed by M SBSs $\mathcal{M} = \{1, ..., M\}$ and N UTs $\mathcal{N} = \{1, ..., N\}$. Each SBS $m \in \mathcal{M}$ is connected to a central scheduler (CS) via a limited backhaul link with capacity c_m, whereas user $n \in \mathcal{N}$ is connected to its serving SBS via a wireless link with capacity $c_{m,n}$. In addition, when deemed feasible, users can establish D2D communications with other users within their communication range[2]. The D2D link capacity between users n and n' is $\ddot{c}_{n,n'}$. The scenario under study is depicted in Fig. 1.

Assume that user n downloads contents from a library of F files, $\mathcal{F} = \{1, ..., F\}$, with probabilities $\mathcal{P}_n = \{p_{n,1}, ..., p_{n,F}\}$. The files in the library $\mathcal{F}$ have lengths of $\mathcal{L} = \{l_1, ..., l_F\}$ respectively, with bitrates $\mathcal{B} = \{b_1, ..., b_F\}$. Further, assume that there are R number of file requests made by users during T time-slots. A request $r \in \mathcal{R} = \{1, ..., R\}$ is served immediately and is said to be *satisfied*, if the rate of delivery is higher than the file bitrate, such that:

$$\frac{l_r}{t'_r - t_r} \geq b_r, \tag{1}$$

where $l_r \in \mathcal{L}$ is the length of the requested file, t_r (t'_r) is the start (end) time of the delivery, respectively, and $b_r \in \mathcal{B}$ is the bitrate of file $f_r \in \mathcal{F}$. Therefore, the file *satisfaction ratio* can be defined as:

$$\eta(\mathcal{R}) = \frac{1}{R} \sum_{r \in \mathcal{R}} \mathbb{1}\left\{ \frac{l_r}{t'_r - t_r} \geq b_r \right\}, \tag{2}$$

where $\mathbb{1}\{...\}$ is the indicator function which returns 1 if the statement holds and 0 otherwise.

The goal of the network operator is to keep this ratio above a target QoS, while reducing the backhaul delivery cost. To achieve this, we closely examine two cases of proactive caching.

A. Backhaul Offloading via Proactive Caching

The importance of the backhaul has increased dramatically over the last couple of years. Indeed, traffic requirements have increased due to the all-IP flat network architecture, thus making backhaul the main network bottleneck.

Let us now suppose that the total backhaul link capacity is less than the total wireless link capacity between SBSs and UTs, such that $\sum_{m \in \mathcal{M}} c_m \ll \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} c_{m,n}$. This assumption stems from the fact that SBSs may not have sufficient high-speed backhaul connections. Since the bottleneck is the backhaul, a smart way of minimizing the backhaul usage is to proactively cache contents at the SBSs, during low-peak demands. Indeed, if the SBSs pre-cache the contents before users' actual requests arrive, corresponding UTs can immediately be served from their SBSs.

Suppose that the backhaul rate during the content delivery for request r at time t is $\lambda_r(t)$. Then, the *backhaul load* can be defined as:

$$\rho(\mathcal{R}) = \frac{1}{R} \sum_{r \in \mathcal{R}} \frac{1}{l_r} \sum_{t=t_r}^{t=t'_r} \lambda_r(t). \tag{3}$$

Further, assume that SBS m has a storage capacity of s_m and the amount of storage usage at time t is $\kappa_m(t)$. Therefore, the backhaul minimization problem subject to backhaul, storage and QoS constraints is written as:

$$\begin{aligned} \underset{t'_r, r \in \mathcal{R}}{\text{minimize}} \quad & \rho(\mathcal{R}) & & \\ \text{subject to} \quad & \lambda_r(t) \leq c_m, & \forall m \in \mathcal{M}, & \\ & \kappa_m(t) \leq s_m, & \forall m \in \mathcal{M}, & \\ & \eta(\mathcal{R}) \geq \eta_{min}, & \forall r \in \mathcal{R}, & \end{aligned} \tag{4}$$

where η_{min} is the minimum target satisfaction ratio. Solving (4) is computationally intractable, and thus, similar to [8] a heuristic solution is used by storing popular files in the caches of SBSs. Here, each SBS m tracks, learns and builds its users' demand profiles to infer on their future requests. Let $\mathbf{P}_m$ denote the discrete file probabilities of users serviced by SBS m, referred to as *popularity matrix* where rows represent users and columns represent file popularities/ratings. A perfect knowledge of $\mathbf{P}_m$ would allow SBSs to precache contents, nevertheless, in practice, this matrix is not perfectly known, large and sparse. Inspired from the *Netflix paradigm* [13] and using supervised machine learning tools, a distributed proactive caching procedure is proposed by exploiting users-files correlations to infer on the probability that user n requests file f.

The proposed caching procedure is composed of a training and prediction step. In the training step, each SBS m builds a model based on the available information of the popularity matrix $\mathbf{P}_m$. This is done by solving a least square minimization problem, in order to calculate the estimated file popularity matrix $\hat{\mathbf{P}}_m$, as follows:

$$\min_{\{b_n, b_f\}} \sum_{n,f} \Big(r_{nf} - \hat{r}_{nf} \Big)^2 + \lambda \Big(\sum_n b_n^2 + \sum_f b_f^2 \Big), \tag{5}$$

where the sum is only over the (n, f) user/file pairs in the training set where user n actually rated file f (i.e., r_{nf}), and the minimization is over all the $N + F$ parameters, where N is the number of users and F the number of files in the training set. In addition, $\hat{r}_{nf} = \bar{r} + b_n + b_f$ is the baseline predictor where b_f models the quality of each file f relative to the average $\bar{r}$ and b_n models the bias of each user n relative to $\bar{r}$. Finally, the weight λ is chosen to balance between regularization and fitting training data.

In the numerical setup, the regularized singular value decomposition (SVD) was chosen for its numerical accuracy (see

[2]D2D communications are assumed to be network-controlled.

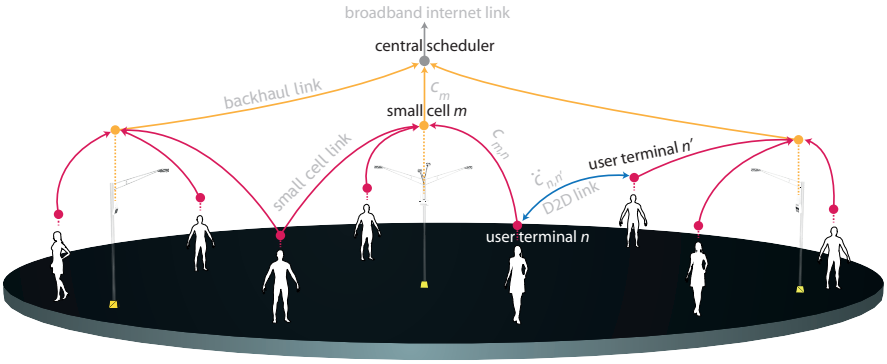


Figure 1: An illustration of the studied network deployment. A central scheduler communicates with M SBSs via backhaul links. In addition to their cellular connections, D2D communication among socially-connected users is depicted.

[14] for comparisons of collaborative filtering (CF) methods). Regularized SVD based CF constructs $\hat{\mathbf{P}}_m$ as the low rank version of $\mathbf{P}_m$. Since the entries of $\mathbf{P}_m$ are partially known, the construction of $\hat{\mathbf{P}}_m$ is done via gradient descent, by exploiting the least-squares property of the singular value decomposition. Subsequently, the proactive caching decision is made based on the estimated file popularity matrix $\hat{\mathbf{P}}_m$.

B. Social-Aware Caching via D2D

Another mean of offloading traffic is by caching contents at the user' cache and harnessing D2D communications for content dissemination. The goal is to reduce the load of the SBSs (and the backhaul load as a consequence). By exploiting the interplay between users' social relationships and physical proximity, each SBS tracks and learns the set of influential users using the social graph. In particular, when a user requests a particular file, the SBS determines whether one of the influential users has the requested file. If so, it directs the influential user to communicate the file to the requesting user via D2D. Otherwise, if the file is not cached by the influential user, the SBS transmits the file directly to the requesting user from the infrastructure network.

Now, assume that UT n has storage capacity $\ddot{s}_n$ and the amount of its storage usage at time t is $\ddot{\kappa}(t)$. Assume also that the total rate of the SBSs during the content delivery of request r at time t is $\dot{\lambda}_r(t)$, and the D2D rate is $\ddot{\lambda}_r(t)$. The *small cell load* can be defined as:

$$\ddot{\rho}(\mathcal{R}) = \frac{1}{R}\sum_{r\in\mathcal{R}}\sum_{t=t_r}^{t=t'_r}\frac{\dot{\lambda}_r(t)}{\dot{\lambda}_r(t)+\ddot{\lambda}_r(t)}. \tag{6}$$

Similar to (4), the D2D caching optimization problem can be formulated as:

$$\begin{aligned}
\underset{t'_r, r\in\mathcal{R}}{\text{minimize}} \quad & \ddot{\rho}(\mathcal{R}) \qquad\qquad\qquad\qquad\qquad\qquad (7)\\
\text{subject to} \quad & \dot{\lambda}_r(t) \le c_{m,n}, \qquad \forall m\in\mathcal{M}, \forall n\in\mathcal{N},\\
& \ddot{\lambda}_r(t) \le \ddot{c}_{n,n'}, \qquad \forall (n,n')\in\mathcal{N},\\
& \ddot{\kappa}_n(t) \le \ddot{s}_n, \qquad \forall n\in\mathcal{N},\\
& \eta(\mathcal{R}) \ge \eta_{min} \qquad \forall r\in\mathcal{R}.
\end{aligned}$$

In order to solve (7), the set of influential users needs to be identified. This can be done by exploiting the social relationships among users via the notion of *centrality* metric [15]. The centrality metric measures the social influence of a node on how well it connects the network, whereby a higher value means a more influential node to its social community. In this work, we use the eigenvector centrality. Let $G=(\mathcal{N},\mathcal{E})$ denote the corresponding social graph composed of N nodes which can be described by its adjacency (or D2D connectivity) matrix $\mathbf{A}_{N\times N}$ with entry $a_{n,n'}$, $n,n'=1,...,N$ equals 1 if link (or edge) $\ddot{c}_{n,n'}$ exists, or 0 otherwise. Let the eigenvalues of $\mathbf{A}$ be $\lambda_1 \ge ... \ge \lambda_N$ in decreasing order and the corresponding eigenvectors be $\mathbf{v}_1,...\mathbf{v}_N$. The eigenvector-centrality is basically the eigenvector $\mathbf{v}_1$ which corresponds to the largest eigenvalue λ_1. A clustering method (i.e., K-means) is then applied for community formation.

Once the set of influential users is identified, the challenge is to disseminate contents within each social community, as a function of users' social ties and physical proximity. Given the large volume of available contents, assume that $\mathcal{F}=\mathcal{F}_0+\mathcal{F}_h$, where $\mathcal{F}_h$ represents the set of contents with viewing histories and $\mathcal{F}_0$ is the set of contents without history. Suppose that each user is interested in one type of available contents $\mathcal{F}$. Let π_f denote the probability that content f is selected by a given user, which by assumption follows a Beta distribution $\beta(\alpha/F,1)$, defined as prior [16]. Therefore, the selection result of user n, defined as the conjugate probability of the Beta distribution follows a Bernoulli distribution. It turns out that the resulting user-file partition is reminiscent to that of the Chinese restaurant process (CRP) [16]. CRP is based upon a metaphor where the objects are customers in a restaurant, and the classes are the tables at which they sit. More precisely, in a restaurant with a large number of tables, each with an infinite number of seats, customers enter the restaurant sequentially, and each one chooses a table at random.

In the CRP with parameter β, each customer chooses an occupied table with a probability proportional to the number of occupants, and chooses the next vacant table with probability proportional to β. Specifically, the first customer chooses the first table with probability $\frac{\beta}{\beta}=1$. The second customer chooses the first table with probability $\frac{1}{1+\beta}$, and the second table with probability $\frac{\beta}{1+\beta}$. After the second customer chooses the second table, the third customer chooses the first table with probability $\frac{1}{2+\beta}$, the second table with

probability $\frac{1}{2+\beta}$ and the third table with probability $\frac{\beta}{2+\beta}$. The process continues until all customers have seats, defining a distribution over allocations of people to tables. Therefore, the decisions of subsequent customers are influenced by the previous customers' feedbacks, in which customers learn from the previous selections to update their beliefs on the files and the probabilities with which they choose their files.

With that in mind, the behaviour of the proactive D2D caching procedure is analogous to the table selection in an CRP. If we view the social network as a Chinese restaurant, the contents as the very large number of files, and the users as the customers, we can interpret the contents dissemination process online by an CRP. That is within every social community, users sequentially request to download their sought-after content, and when a user downloads its content, the recorded hits are recorded (i.e., history). In turn, this action affects the probability that this content will be requested by others users within the same social community, where popular contents are requested more frequently and new contents less frequently. Let $\mathbf{Z}_{N\times F}$ be a random binary matrix indicating which contents are selected by each user, where $z_{nf} = 1$ if user n selects content f and 0 otherwise. It can be shown that [16]:

$$P(\mathbf{Z}) = \frac{\beta^{F'}\Gamma(\beta)}{\Gamma(\beta+N)}\prod_{f=1}^{F'}(m_f-1)! \tag{8}$$

in which $\Gamma(.)$ is the Gamma function [17], m_f is the number of users currently assigned to content f (i.e., viewing history) and F' is the number of partitions with $m_f > 0$.

III. Numerical results

In this section, we evaluate the performance of proactive caching and provide key insights under two different scenarios.

A. Backhaul Offloading via Proactive Caching

The parameters for the numerical setup are given in Table I. For simplification, the link and storage capacities are assumed to be equal. Three regimes of interest are considered: (i) low load, (ii) medium load, and (iii) high load.

Over a time duration T, R numbers of requests are generated. The arrival times of user requests are drawn uniformly at random, and the requested files are sampled from the ZipF(α) distribution. At $t = 0$, the popularity matrix is constructed perfectly. Out of 20% of the elements of this matrix are removed uniformly at random and the remaining elements of the matrix are used for training in CF. These removed entries are then predicted using the regularized SVD [18]. After the popularity matrix estimation, proactive caching is applied by storing the most popular files subject to the SBSs' storage constraints. Having these files locally in the cache of SBSs and starting from $t = 0$, the delivery is carried out by each SBS until all requests are served. Random caching is used as a baseline procedure, referred to as *reactive*.

Three parameters of interests are considered for the performance plots of both proactive and reactive caching approaches: (i) number of requests R, (ii) total cache size S, and (iii) ZipF distribution parameter α. To see the percentages of differences between the proactive and reactive approaches, plots are normalized. The evolution of the satisfaction ratios and backhaul loads are shown in Fig. 2. Each subfigure represents the impact of one parameter for a given regime while the other parameters are kept fixed.

Parameter	Description	Value
T	Time slots	1024 seconds
M	Number of small cells	4
N	Number of user terminals	32
F	Number of files	128
l_f	Length of file f	1 Mbit
b_f	Bitrate of file f	1 Mbit/s
$\sum_m c_m$	Total backhaul link capacity	2 Mbit/s
$\sum_m \sum_n c_{m,n}$	Total wireless link capacity	64 Mbit/s
R	Number of requests	$0 \sim 2048$
S	Total cache size	$0 \sim l_f \times F$
α	ZipF parameter	$0 \sim 2$

Table I: List of parameters for the numerical setup of the proactive small cell networks.

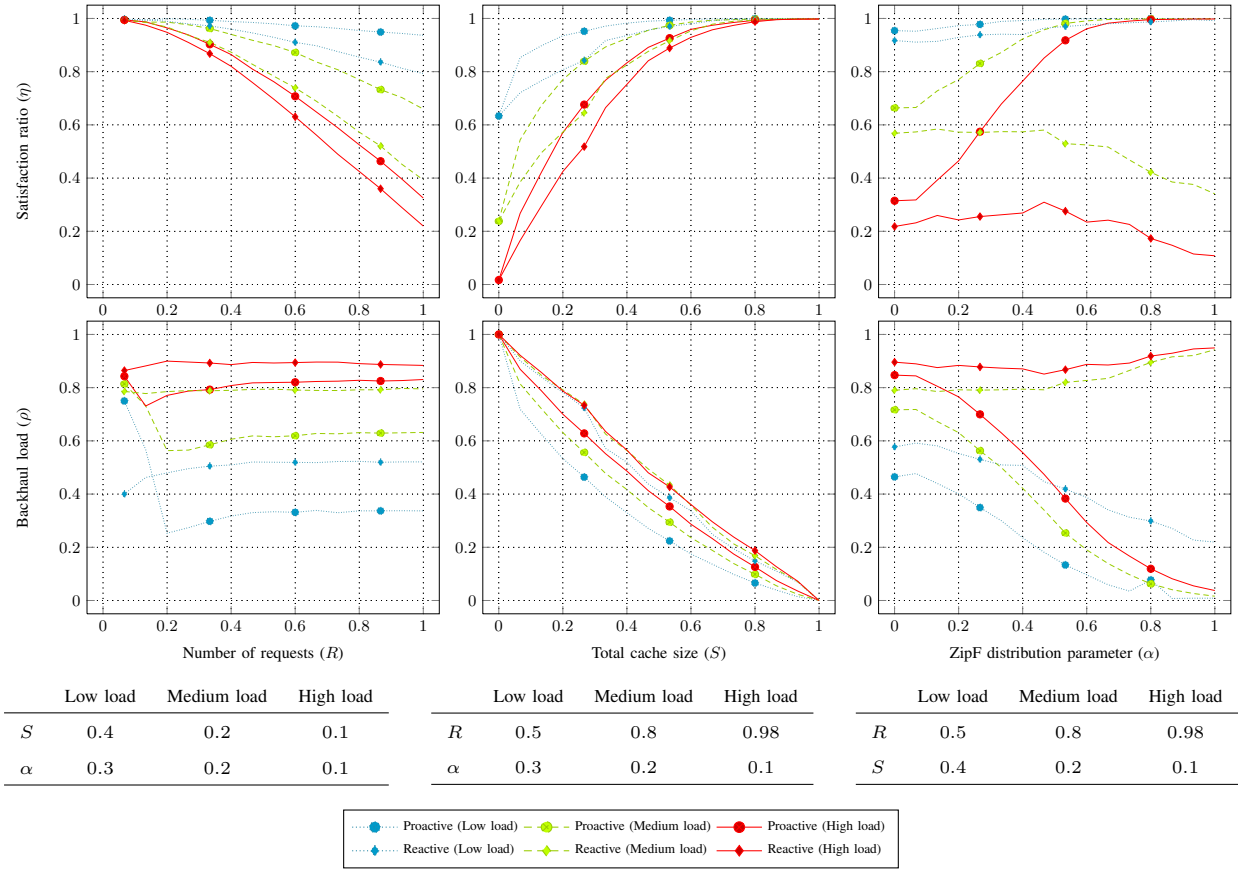
1) Impact of number of requests: The satisfaction ratio decreases with the increase in users' requests. This is evident as the amount of capacity and storage resources are limited. However, the proactive caching outperforms the reactive approach in terms of satisfaction ratio. On the other hand, the reactive approach generates less load on the backhaul in the case of very small number of requests. This situation can be explained by the *cold start* phenomenon where the CF cannot draw any inference due to non-sufficient amount of information about the popularity matrix. Therefore the random caching for a fixed library size outperforms the proposed approach at low load. However, as users' requests increase, the proactive approach minimizes the backhaul load outperforming the reactive approach, after which the gains level off.

2) Impact of cache size: It can be seen that as the total storage size of small cell base stations increases, the satisfaction ratio approaches 1 and the backhaul load tends to 0. This cannot be easily achieved in practice as it requires storing all file requests whereas SBSs have limited storage. Therefore, for reasonable values of cache size, it can be seen that the proactive caching outperforms the reactive case in terms of satisfaction ratio and backhaul load.

3) Impact of popularity distribution: As the popularity of some files increases as compared to others (i.e., α increases), the gain of the proactive caching becomes higher compared to the random approach in all regimes. Going from the low load regime to the high load regime, the gains further improve.

B. Social-Aware Caching via D2D

To see the impact of the parameters of interest, wireless link capacities are equally divided among users. The total D2D link capacity is shared among users according to their social links. The parameters used in the numerical setup are summarized in Table II.



	Low load	Medium load	High load
S	0.4	0.2	0.1
α	0.3	0.2	0.1

	Low load	Medium load	High load
R	0.5	0.8	0.98
α	0.3	0.2	0.1

	Low load	Medium load	High load
R	0.5	0.8	0.98
S	0.4	0.2	0.1

Figure 2: Backhaul Offloading via Proactive Caching: Dynamics of the satisfied requests and backhaul load with respect to the number of requests, total cache size and ZipF parameter.

Parameter	**Description**	**Value**
T	Time slots	1024 seconds
M	Number of small cells	4
K	Number of communities	3
N	Number of user terminals	32
F	Number of files	128
l_f	Length of file f	1 Mbit
b_f	Bitrate of file f	1 Mbit/s
$\sum_m \sum_n c_{m,n}$	Total SBSs link capacity	32 Mbit/s
$\sum_n \sum_{n',n' \neq n} \ddot{c}_{n,n'}$	Total D2D link capacity	64 Mbit/s
R	Number of requests	$0 \sim 9464$
S	Total D2D cache size	$0 \sim l_f \times F$
β	CRP parameter	$0 \sim 100$

Table II: List of parameters for the numerical setup of the social networks aware caching via D2D.

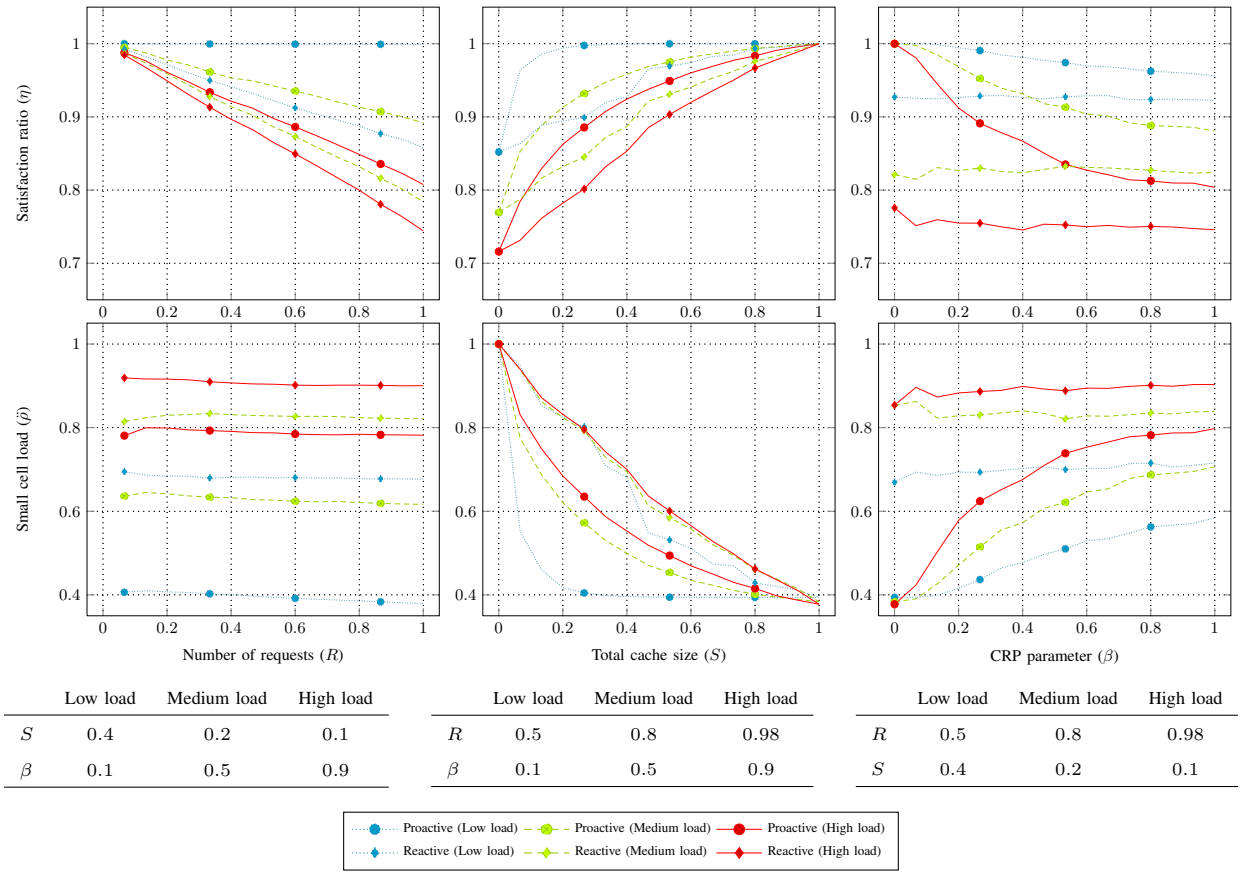
At $t = 0$, user selection and their requests' arrival times are sampled uniformly at random for a time interval T. The social graph is synthetically generated by using the preferential attachment model [19]. The eigenvector centrality is used to infer on the set of influential users in the social network, where users are formed into K clusters via K-means clustering [20]. Within every social community, the file popularity distribution is sampled from the CRP(β) and the proactive caching is carried out by storing popular files within each community. Similarly, random caching is used as a baseline.

Three parameters are of interest: (i) number of requests R, total D2D cache size S and CRP concentration parameter β. The results are normalized for ease of comparison. As in the previous case, similar evaluation metrics are used. The evolution of the satisfaction ratio and small cell load with respect to these parameters are plotted in Fig. 3. In the following, we discuss the impact of these parameters:

1) Impact of number of requests: As the number of requests increases, the satisfaction ratio decreases rapidly, whereas the small cell load decreases at a low pace. It can be clearly seen that the proactive caching approach outperforms the reactive approach in all regimes.

2) Impact of D2D cache size: As the D2D cache size increases, it can be seen that the satisfaction ratio increases while the the small cell load decreases. Moreover, the increment in cache size improves the performance of the reactive approach, however the gains of the proactive caching approach are higher.

3) Impact of CRP concentration parameter: In the case of an increment in β (i.e., the number of files grows), the satisfaction ratio and the small cell load tend to be constant under the reactive approach. On the other hand, the proactive



	Low load	Medium load	High load
S	0.4	0.2	0.1
β	0.1	0.5	0.9

	Low load	Medium load	High load
R	0.5	0.8	0.98
β	0.1	0.5	0.9

	Low load	Medium load	High load
R	0.5	0.8	0.98
S	0.4	0.2	0.1

Figure 3: Social-Aware Caching via D2D: Dynamics of the satisfied requests and small cell load with respect to the number of requests, total cache size and CRP concentration parameter β.

caching approach exhibits a better performance, and the gap between the proactive and reactive approaches gets smaller as β increases. This is a by-product of the increasing file library size with a fixed cache size.

IV. Conclusion

In this paper, we studied a novel proactive networking paradigm where caching plays an important role. The proactive caching solution exploits users' predictable demands, storage, and their social relationships to minimize peak mobile data traffic demands. It was demonstrated that precaching strategic contents at the network edge engenders significant backhaul offloading gains and resource savings. Our future work will explore distributed MIMO caching and multicast caching.

References

[1] E. Baştuğ, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5g wireless networks," *Submitted*, 2013.

[2] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2012-2017," *White Paper, [Online] http://goo.gl/uQ0DJQ*, 2013.

[3] Ericsson, "5g radio access - research and vision," *White Paper, [Online] http://goo.gl/Huf0b6*, 2012.

[4] J. Hoydis, M. Kobayashi, and M. Debbah, "Green small-cell networks," *IEEE Vehicular Technology Magazine*, vol. 6(1), pp. 37–43, 2011.

[5] J. Andrews, "Seven ways that hetnets are a cellular paradigm shift," *IEEE Communications Magazine*, vol. 51, no. 3, pp. 136–144, 2013.

[6] M. Bennis, M. Simsek, W. Saad, S. Valentin, M. Debbah, and A. Czylwik, "When cellular meets wifi in wireless small cell networks," *IEEE Communication Magazine, Special Issue in HetNets*, June 2013.

[7] Intel, "Rethinking the small cell business model," *White Paper, [Online] http://goo.gl/c2r9jX*, 2012.

[8] E. Baştuğ, J.-L. Guénégo, and M. Debbah, "Proactive small cell networks," in *20th International Conference on Telecommunications (ICT)*, Casablanca, Morocco, May 2013.

[9] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 1107–1115.

[10] J. Tadrous, A. Eryilmaz, and H. E. Gamal, "Proactive data download and user demand shaping for data networks," *submitted to IEEE Transactions on Information Theory, [Online] arXiv: 1304.5745*, 2013.

[11] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of distributed caching in d2d wireless networks," *[Online] arXiv: 1304.5856*, 2013.

[12] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[13] Netflix, "Netflix prize," *[Online] http://www.netflixprize.com*, 2009.

[14] J. Lee, M. Sun, and G. Lebanon, "A comparative study of collaborative filtering algorithms," *[Online] arXiv: 1205.3193*, 2012.

[15] M. Newman, *Networks: an introduction*. Oxford University Press, 2009.

[16] T. L. Griffiths and Z. Ghahramani, "The indian buffet process: An introduction and review," *J. Mach. Learn. Res.*, vol. 12, pp. 1185–1224, Jul. 2011.

[17] M. A. I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. National Bureau of Standards Applied Mathematics Series 55. Tenth Printing, 1972.

[18] P. Arkadiusz, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and workshop Vol. 2007.*, 2007.

[19] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[20] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651 – 666, 2010.